



Atty. Docket No.: 30390-14

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of:

Ken UMENO

Serial No.: 10/803,587

Group Art Unit: Not Assigned

Filed: 17 March 2004

Examiner: Not Assigned

For: RANDOM SEQUENCE
GENERATING APPARATUS,
ENCRYPTION/DECRYPTION
APPARATUS, RANDOM
SEQUENCE GENERATING
METHOD, ENCRYPTION/
DECRYPTION METHOD AND
PROGRAM

San Diego, California

March 26, 2004

Mail Stop: PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

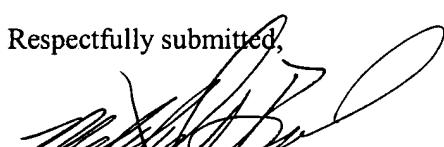
Dear Sir or Madam:

Transmitted herewith is Priority Document: Japanese Application Serial No. JP 2003-075438. Although it is believed that no fees are due for this submission, the Commissioner is authorized to charge any shortage in fees due in connection with the filing of this paper to our Deposit Account No. 50-2298 in the name of Luce, Forward, Hamilton & Scripps LLP.

Respectfully submitted,

3/26/04

Date



Mitchell P. Brook
Attorney for Applicant(s)
Reg. No. 32,967
c/o LUCE, FORWARD, HAMILTON
& SCRIPPS LLP
11988 El Camino Real, Suite 200
San Diego, California 92130
Telephone No.: (858) 720-6300

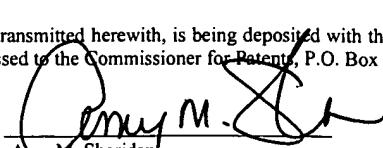
CERTIFICATE OF MAILING

I hereby certify that this correspondence, and anything referred to as transmitted herewith, is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Date: 26 Mar. 2004

By:

Amy M. Sheridan



日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 3月19日
Date of Application:

出願番号 特願2003-075438
Application Number:

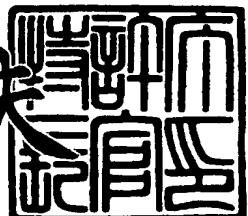
[ST. 10/C] : [JP2003-075438]

出願人 独立行政法人通信総合研究所
Applicant(s):

2004年 2月17日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



【書類名】 特許願
【整理番号】 CRL-02-43
【提出日】 平成15年 3月19日
【あて先】 特許庁長官 殿
【国際特許分類】 H04B 1/00
【発明者】
【住所又は居所】 東京都小金井市貫井北町4-2-1 独立行政法人通信
総合研究所内
【氏名】 梅野 健
【特許出願人】
【識別番号】 301022471
【氏名又は名称】 独立行政法人通信総合研究所
【代理人】
【識別番号】 100095407
【弁理士】
【氏名又は名称】 木村 満
【選任した代理人】
【識別番号】 100110135
【弁理士】
【氏名又は名称】 石井 裕一郎
【提出物件の目録】
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 0112098
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 亂数列生成装置、暗号化復号化装置、乱数列生成方法、暗号化復号化方法、ならびに、プログラム

【特許請求の範囲】

【請求項 1】

wビットの乱数の列を生成する乱数列生成装置であって、種受付部と、初期化部と、変換部と、回転部と、更新部と、出力部と、を備え、整数n, m (1≤n≤m-1)に対して、

前記種受付部は、wビットの整数の列s₁, s₂, …, s_n, …, s_mを種として受け付け、
前記初期化部は、当該受け付けられた整数の列s₁, s₂, …, s_n, …, s_mを、整数列x₁, x₂, …, x_n, …, x_mとして前記変換部に与え、

前記変換部は、当該与えられた整数列x₁, x₂, …, x_n, …, x_mのそれぞれに対して所定の変換を施してwビットの整数の列y₁, y₂, …, y_n, …, y_mを得て、

前記回転部は、当該整数の列y_{n+1}, …, y_mから回転ビット数を得て、当該整数の列y₁, y₂, …, y_n, …, y_mをwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列z₁, z₂, …, z_n, …, z_mを得て、

前記更新部は、当該整数の列z₁, z₂, …, z_n, …, z_mを整数列x₁, x₂, …, x_n, …, x_mとして前記変換部に与え、

前記出力部は、前記変換部における変換ならびに前記回転部における回転が、所定の回数繰り返された場合、最後に得られたz₁, z₂, …, z_nを、乱数列r₁, r₂, …, r_nとして出力する

ことを特徴とするもの。

【請求項 2】

請求項 1 に記載の乱数列生成装置であって、

前記変換部は、写像g(・, ・)を用いて、整数i (1≤i≤m-1)についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とするもの。

【請求項3】

請求項1に記載の乱数列生成装置であって、

前記変換部は、所定の整数cと、写像 $g(\cdot, \cdot)$ と、を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行う

ことを特徴とするもの。

【請求項4】

請求項1に記載の乱数列生成装置であって、

前記変換部は、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とするもの。

【請求項5】

請求項2から4のいずれか1項に記載の乱数列生成装置であって、

当該写像 $g(\cdot, \cdot)$ は、所定の写像 $h(\cdot)$ と所定の整数 q ($0 \leq q \leq 2^w-1$)により

$$g(a, b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義される

ことを特徴とするもの。

【請求項6】

請求項5に記載の乱数列生成装置であって、

当該写像 $h(\cdot)$ は、

$$h(a) = a$$

と定義される

ことを特徴とするもの。

【請求項7】

請求項 5 に記載の乱数列生成装置であって、
当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットをクリアする演算
により定義される
ことを特徴とするもの。

【請求項 8】

請求項 5 に記載の乱数列生成装置であって、
当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットを反転する演算に
より定義される
ことを特徴とするもの。

【請求項 9】

請求項 5 に記載の乱数列生成装置であって、
当該写像 $h(\cdot)$ は、与えられた値の数値表現の最下位 2 ビットを 0 1 に置換す
る演算により定義される
ことを特徴とするもの。

【請求項 10】

請求項 1 から 9 のいずれか 1 項に記載の乱数列生成装置であって、
前記回転部は、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、
これから所定の位置のビットを 1 個以上抽出して並べたビット列を整数と見たと
きの整数値を、回転ビット数として得る
ことを特徴とするもの。

【請求項 11】

請求項 10 に記載の乱数列生成装置であって、
前記回転部は、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、
これから所定の位置のビットの値により、回転する方向を決める
ことを特徴とするもの。

【請求項 12】

請求項 1 から 11 のいずれか 1 項に記載の乱数生成装置であって、
前記回転部は、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の
列 y_1, y_2, \dots, y_n を w_n ビットのビット列と見て、これに対して当該得られた回転ビ

ット数の回転演算を行って、得られた w_n ビットのビット列から w ビットの整数の列 z_1, z_2, \dots, z_n を得て、整数の列 y_{n+1}, \dots, y_m を $w(m-n)$ ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた $w(m-n)$ ビットのビット列から w ビットの整数の列 z_{n+1}, \dots, z_m を得る

ことを特徴とするもの。

【請求項 1 3】

乱数生成部と、メッセージ受付部と、暗号化復号化部と、を備える暗号化復号化装置であって、

前記乱数生成部は、請求項 1 から 1 2 のいずれか 1 項に記載の乱数生成装置により、乱数列 r_1, r_2, \dots, r_n を生成し、

前記メッセージ受付部は、 w ビットの整数の整数列 p_1, p_2, \dots をメッセージとして受け付け、

前記暗号化復号化部は、 w ビットの整数の列 $p_1 \text{ xor } r_1, p_2 \text{ xor } r_2, \dots, p_i \text{ xor } r_{((i+n-1) \bmod n) + 1}, \dots$ を暗号化もしくは復号化の結果として出力する

ことを特徴とするもの。

【請求項 1 4】

w ビットの乱数の列を生成する乱数列生成方法であって、種受付工程と、初期化工程と、変換工程と、回転工程と、更新工程と、出力工程と、を備え、整数 n, m ($1 \leq n \leq m-1$) に対して、

前記種受付工程では、 w ビットの整数の列 $s_1, s_2, \dots, s_n, \dots, s_m$ を種として受け付け、

前記初期化工程では、当該受け付けられた整数の列 $s_1, s_2, \dots, s_n, \dots, s_m$ を、整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として前記変換工程に与え、

前記変換工程では、当該与えられた整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ のそれぞれに対して所定の変換を施して w ビットの整数の列 $y_1, y_2, \dots, y_n, \dots, y_m$ を得て、

前記回転工程では、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の列 $y_1, y_2, \dots, y_n, \dots, y_m$ を wm ビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られた wm ビットのビット列から w ビットの整数の列 $z_1, z_2, \dots, z_n, \dots, z_m$ を得て、

前記更新工程では、当該整数の列 $z_1, z_2, \dots, z_n, \dots, z_m$ を整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として前記変換工程に与え、

前記出力工程では、前記変換工程における変換ならびに前記回転工程における回転が、所定の回数繰り返された場合、最後に得られた z_1, z_2, \dots, z_n を、乱数列 r_1, r_2, \dots, r_n として出力する
ことを特徴とする方法。

【請求項 1 5】

請求項 1 4 に記載の乱数列生成方法であって、

前記変換工程では、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項 1 6】

請求項 1 4 に記載の乱数列生成方法であって、

前記変換工程では、所定の整数 c と、写像 $g(\cdot, \cdot)$ と、を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項 1 7】

請求項 1 4 に記載の乱数列生成方法であって、

前記変換工程では、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う

ことを特徴とする方法。

【請求項18】

請求項15から17のいずれか1項に記載の乱数列生成方法であって、

当該写像 $g(\cdot, \cdot)$ は、所定の写像 $h(\cdot)$ と所定の整数 q ($0 \leq q \leq 2^w - 1$)により

$$g(a, b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義される

ことを特徴とする方法。

【請求項19】

請求項18に記載の乱数列生成方法であって、

当該写像 $h(\cdot)$ は、

$$h(a) = a$$

と定義される

ことを特徴とする方法。

【請求項20】

請求項18に記載の乱数列生成方法であって、

当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットをクリアする演算

により定義される

ことを特徴とする方法。

【請求項21】

請求項18に記載の乱数列生成方法であって、

当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットを反転する演算に

より定義される

ことを特徴とする方法。

【請求項22】

請求項18に記載の乱数列生成方法であって、

当該写像 $h(\cdot)$ は、与えられた値の数値表現の最下位2ビットを01に置換す

る演算により定義される

ことを特徴とする方法。

【請求項23】

請求項14から22のいずれか1項に記載の乱数列生成方法であって、前記回転工程では、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットを1個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得ることを特徴とする方法。

【請求項24】

請求項23に記載の乱数列生成方法であって、前記回転工程では、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決める

ことを特徴とする方法。

【請求項25】

請求項14から24のいずれか1項に記載の乱数生成方法であって、前記回転工程では、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の列 y_1, y_2, \dots, y_n を wn ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた wn ビットのビット列から w ビットの整数の列 z_1, z_2, \dots, z_n を得て、整数の列 y_{n+1}, \dots, y_m を $w(m-n)$ ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた $w(m-n)$ ビットのビット列から w ビットの整数の列 z_{n+1}, \dots, z_m を得ることを特徴とする方法。

【請求項26】

乱数生成工程と、メッセージ受付工程と、暗号化復号化工程と、を備える暗号化復号化方法であって、

前記乱数生成工程では、請求項13から23のいずれか1項に記載の乱数生成方法により、乱数列 r_1, r_2, \dots, r_n を生成し、

前記メッセージ受付工程では、 w ビットの整数の整数列 p_1, p_2, \dots をメッセージとして受け付け、

前記暗号化復号化工程では、 w ビットの整数の列 $p_1 \text{ xor } r_1, p_2 \text{ xor } r_2, \dots, p_i \text{ xor } r((i+n-1) \bmod n) + 1, \dots$ を暗号化もしくは復号化の結果として出力することを特徴とする方法。

【請求項27】

コンピュータを、請求項1から12のいずれか1項に記載の乱数列生成装置として機能させることを特徴とするプログラム。

【請求項28】

コンピュータを、請求項13に記載の暗号化復号化装置として機能させることを特徴とするプログラム。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は、乱数列生成装置、暗号化復号化装置、乱数列生成方法、暗号化復号化方法、ならびに、プログラムに関する。

【0002】**【従来の技術】**

従来から、種々の乱数列生成のための技術が提案されている。これらの技術によって得られた乱数は、たとえば、モンテカルロ法による各種の物理現象、化学現象などの模擬実験や、秘密通信のブロック暗号システムにおいて用いられる。

【0003】**【発明が解決しようとする課題】**

さて、このような乱数列の生成技術においては、得られた乱数列に含まれる数値の分布ができるだけ一様であることや、当該数値のコンピュータにおける数値表現の所定のビットのみを見た場合に、当該ビットの「0」と「1」の出現頻度にできるだけ偏りがないことや、乱数列の周期ができるだけ長いことなど、種々の性質を満たすことが望ましい。

【0004】

本発明は、生成される乱数列が乱数列として好ましい性質を有するような乱数列生成装置、乱数列生成方法、ならびに、これらを用いた暗号化復号化装置、暗号化復号化方法、ならびに、これらをコンピュータによって実現するためのプログラムを提供することを目的とする。

【0005】

【課題を解決するための手段】

以上の目的を達成するため、本発明の原理にしたがって、下記の発明を開示する。

【0006】

本発明の第1の観点に係る乱数列生成装置は、wビットの乱数の列を生成し、種受付部と、初期化部と、変換部と、回転部と、更新部と、出力部と、を備え、整数n, m (1≤n≤m-1) に対して、以下のように構成する。

【0007】

すなわち、種受付部は、wビットの整数の列s₁, s₂, …, s_n, …, s_mを種として受け付ける。

【0008】

一方、初期化部は、当該受け付けられた整数の列s₁, s₂, …, s_n, …, s_mを、整数列x₁, x₂, …, x_n, …, x_mとして変換部に与える。

【0009】

さらに、変換部は、当該与えられた整数列x₁, x₂, …, x_n, …, x_mのそれぞれに対して所定の変換を施してwビットの整数の列y₁, y₂, …, y_n, …, y_mを得る。

【0010】

そして、回転部は、当該整数の列y_{n+1}, …, y_mから回転ビット数を得て、当該整数の列y₁, y₂, …, y_n, …, y_mをwmビットのビット列と見て、これ、もしくは、これの一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列z₁, z₂, …, z_n, …, z_mを得る。

【0011】

一方、更新部は、当該整数の列z₁, z₂, …, z_n, …, z_mを整数列x₁, x₂, …, x_n, …, x_mとして変換部に与える。

【0012】

さらに、出力部は、変換部における変換ならびに回転部における回転が、所定の回数繰り返された場合、最後に得られたz₁, z₂, …, z_nを、乱数列r₁, r₂, …, r_nとして出力する。

【0013】

また、本発明の乱数列生成装置において、変換部は、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【0014】

また、本発明の乱数列生成装置において、変換部は、所定の整数 c と、写像 $g(\cdot, \cdot)$ と、を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行うように構成することができる。

【0015】

また、本発明の乱数列生成装置において、変換部は、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【0016】

また、本発明の乱数列生成装置において、当該写像 $g(\cdot, \cdot)$ は、所定の写像 $h(\cdot)$ と所定の整数 q ($0 \leq q \leq 2^w - 1$)により

$$g(a, b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義されるように構成することができる。

【0017】

また、本発明の乱数列生成装置において、当該写像 $h(\cdot)$ は、

$$h(a) = a$$

と定義されるように構成することができる。

【0018】

また、本発明の乱数列生成装置において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットをクリアする演算により定義されるように構成すること

ができる。

【0019】

また、本発明の乱数列生成装置において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットを反転する演算により定義されるように構成することができる。

【0020】

また、本発明の乱数列生成装置において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の最下位2ビットを01に置換する演算により定義されるように構成することができる。

【0021】

また、本発明の乱数列生成装置において、回転部は、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットを1個以上抽出して並べたビット列を整数と見たときの整数值を、回転ビット数として得るよう構成することができる。

【0022】

また、本発明の乱数列生成装置において、回転部は、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決めるよう構成することができる。

【0023】

また、本発明の乱数列生成装置において、回転部は、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の列 y_1, y_2, \dots, y_n を w_n ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた w_n ビットのビット列から w ビットの整数の列 z_1, z_2, \dots, z_n を得て、整数の列 y_{n+1}, \dots, y_m を $w(m-n)$ ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた $w(m-n)$ ビットのビット列から w ビットの整数の列 z_{n+1}, \dots, z_m を得るように構成することができる。すなわち、 z_i は、 u_i を所定の回転ビット数だけ回転演算したものである。

【0024】

本発明の他の観点に係る暗号化復号化装置は、乱数生成部と、メッセージ受付

部と、暗号化復号化部と、を備え、以下のように構成する。

【0025】

すなわち、乱数生成部は、上記の乱数生成装置により、乱数列 r_1, r_2, \dots, r_n を生成する。

【0026】

一方、メッセージ受付部は、wビットの整数の整数列 p_1, p_2, \dots をメッセージとして受け付ける。

【0027】

さらに、暗号化復号化部は、wビットの整数の列 $p_1 \text{ xor } r_1, p_2 \text{ xor } r_2, \dots, p_i \text{ xor } r((i+n-1) \bmod n) + 1, \dots$ を暗号化もしくは復号化の結果として出力する。

【0028】

本発明の他の観点に係る乱数列生成方法は、wビットの乱数の列を生成し、種受付工程と、初期化工程と、変換工程と、回転工程と、更新工程と、出力工程と、を備え、整数 n, m ($1 \leq n \leq m-1$)に対して、以下のように構成する。

【0029】

すなわち、種受付工程では、wビットの整数の列 $s_1, s_2, \dots, s_n, \dots, s_m$ を種として受け付ける。

【0030】

一方、初期化工程では、当該受け付けられた整数の列 $s_1, s_2, \dots, s_n, \dots, s_m$ を、整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として変換工程に与える。

【0031】

さらに、変換工程では、当該与えられた整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ のそれぞれに対して所定の変換を施してwビットの整数の列 $y_1, y_2, \dots, y_n, \dots, y_m$ を得る。

【0032】

そして、回転工程では、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の列 $y_1, y_2, \dots, y_n, \dots, y_m$ をwmビットのビット列として見て、これ、もしくは、この一部に対して当該得られた回転ビット数の回転演算を行って、得られたwmビットのビット列からwビットの整数の列 $z_1, z_2, \dots, z_n, \dots, z_m$ を得る。

【0033】

一方、更新工程では、当該整数の列 $z_1, z_2, \dots, z_n, \dots, z_m$ を整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として変換工程に与える。

【0034】

さらに、出力工程では、変換工程における変換ならびに回転工程における回転が、所定の回数繰り返された場合、最後に得られた z_1, z_2, \dots, z_n を、乱数列 r_1, r_2, \dots, r_n として出力する。

【0035】

また、本発明の乱数列生成方法において、変換工程では、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【0036】

また、本発明の乱数列生成方法において、変換工程では、所定の整数 c と、写像 $g(\cdot, \cdot)$ と、を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行うように構成することができる。

【0037】

また、本発明の乱数列生成方法において、変換工程では、写像 $g(\cdot, \cdot)$ を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行うように構成することができる。

【0038】

また、本発明の乱数列生成方法において、当該写像 $g(\cdot, \cdot)$ は、所定の写像 $h(\cdot)$ と所定の整数 q ($0 \leq q \leq 2^w - 1$)により

$$g(a, b) = 2b^2 + h(a)b + q \pmod{2^w}$$

と定義されるように構成することができる。

【0039】

また、本発明の乱数列生成方法において、当該写像 $h(\cdot)$ は、

$$h(a) = a$$

と定義されるように構成することができる。

【0040】

また、本発明の乱数列生成方法において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットをクリアする演算により定義されるように構成することができる。

【0041】

また、本発明の乱数列生成方法において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の所定のビットを反転する演算により定義されるように構成することができる。

【0042】

また、本発明の乱数列生成方法において、当該写像 $h(\cdot)$ は、与えられた値の数値表現の最下位2ビットを01に置換する演算により定義されるように構成することができる。

【0043】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットを1個以上抽出して並べたビット列を整数と見たときの整数値を、回転ビット数として得るように構成することができる。

【0044】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列 y_{n+1}, \dots, y_m を $w(m-n+1)$ ビットのビット列と見て、これから所定の位置のビットの値により、回転する方向を決めるように構成することができる。

【0045】

また、本発明の乱数列生成方法において、回転工程では、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得て、当該整数の列 y_1, y_2, \dots, y_n を w_n ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた

wn ビットのビット列から w ビットの整数の列 z_1, z_2, \dots, z_n を得て、整数の列 y_{n+1}, \dots, y_m を $w(m-n)$ ビットのビット列と見て、これに対して当該得られた回転ビット数の回転演算を行って、得られた $w(m-n)$ ビットのビット列から w ビットの整数の列 z_{n+1}, \dots, z_m を得るように構成することができる。すなわち、 z_i は、 u_i を所定の回転ビット数だけ回転演算したものである。

【0046】

本発明の他の観点に係る暗号化復号化方法は、乱数生成工程と、メッセージ受付工程と、暗号化復号化工程と、を備え、以下のように構成する。

【0047】

すなわち、乱数生成工程では、上記の乱数生成方法により、乱数列 r_1, r_2, \dots, r_n を生成する。

【0048】

一方、メッセージ受付工程では、 w ビットの整数の整数列 p_1, p_2, \dots をメッセージとして受け付ける。

【0049】

さらに、暗号化復号化工程では、 w ビットの整数の列 $p_1 \text{ xor } r_1, p_2 \text{ xor } r_2, \dots, p_i \text{ xor } r((i+n-1) \bmod n) + 1, \dots$ を暗号化もしくは復号化の結果として出力する。

【0050】

本発明の他の観点に係るプログラムは、コンピュータを、上記の乱数列生成装置もしくは暗号化復号化装置として機能させ、もしくは、コンピュータに、上記の乱数列生成方法もしくは暗号化復号化方法を実行させるように構成する。

【0051】

これらのプログラムは、コンパクトディスク、フレキシブルディスク、ハードディスク、光磁気ディスク、デジタルビデオディスク、磁気テープ、半導体メモリ等のコンピュータ読取可能な情報記録媒体に記録することができる。

【0052】

上記プログラムは、当該プログラムが実行されるコンピュータとは独立して、コンピュータ通信網を介して配布・販売することができる。また、上記情報記録

媒体は、当該コンピュータとは独立して配布・販売することができる。

【0053】

【発明の実施の形態】

以下に本発明の実施形態を説明する。なお、以下に説明する実施形態は説明のためのものであり、本願発明の範囲を制限するものではない。したがって、当業者であればこれらの各要素もしくは全要素をこれと均等なものに置換した実施形態を採用することが可能であるが、これらの実施形態も本願発明の範囲に含まれる。

【0054】

(発明の実施の形態)

以下で説明する本発明の実施形態においては、「 w ビットの数値表現による乱数」の列を生成するために、有限体上の非線型変換として、所定の写像 $h(\cdot)$ と所定の整数 q ($0 \leq q \leq 2^w - 1$)とを用いて

$$g(a, b) = 2b^2 + h(a)b + q \pmod{2^w}$$

により定義される写像 $g(\cdot, \cdot)$ を用いる。

【0055】

写像 $h(\cdot)$ としては、恒等写像

$$h(a) = a$$

を用いることができる。

【0056】

また、所定のマスク値MASKを利用することにより、与えられた値 a の数値表現の所定のビットをクリアする演算

$$h(a) = a \text{ and MASK}$$

や、所定のビットを反転する演算

$$h(a) = a \text{ xor MASK}$$

を採用してもよい。

【0057】

さらに、最下位の2ビットを値01に変換する演算

$$h(a) = (a \text{ and } (\text{not } 3)) \text{ or } 1$$

などを採用することができる。

【0058】

ここで、各演算子は値aの数値表現（整数表現）に対するもので、andはビット積（ビットアンド）、xorはビット排他的和（ビットエクスクルーシブオア）、notはビット反転（ビットノット）、orはビット和（ビットオア）にそれぞれ相当するものである。

【0059】

したがって、これらの演算は、コンピュータにおいては桁上がりや桁下がりなどをとりたてて考慮せずに、wビットの整数演算として用意されているものをそのまま適用して実現することができる。

【0060】

またwの値は、当該コンピュータのC P U (Central Processing Unit) に用意されている機械語のビット幅、もしくは、これよりも小さい幅に対応させることが望ましい。

【0061】

現在のところ、世界最高速のブロック暗号技術といわれているR C 6は、有限体上の非線型変換

$$f(x) = 2x^2 + x \pmod{2^w}$$

を用いることによって実現されており、これによって、ある種から生成される乱数列は、これとは異なる種から生成される乱数列とは常に異なるものであり（1対1性）、生成される乱数列の最長周期が 2^w-1 となっている。

【0062】

本実施形態において採用される写像 $g(\cdot, \cdot)$ は、R C 6における有限体上の非線型変換をさらに一般化したものであり、

$$h(a) = 1;$$

$$q = 0$$

とした $g(\cdot, \cdot)$ を採用した場合には、R C 6と同等の乱数列の生成能力を有する。なお、本発明においては、上記のR C 6同等の写像の以外の写像を選択できるため、さまざまなバリエーションの乱数を得ることができる。

【0063】

また、これ以外の演算や値を選択した場合にも、良好な乱数列が得られることが、実験により実証されている。

【0064】

図1は、本実施形態に係る乱数生成装置の概要構成を示す模式図である。図2は、本実施形態に係る乱数生成装置において実行させる処理の制御の流れを示すフローチャートである。以下、これらの図を参照して、本実施形態について、詳細に説明する。

【0065】

乱数列生成装置101は、wビットの乱数の列を生成し、種受付部102と、初期化部103と、変換部104と、回転部105と、更新部106と、出力部107と、を備える。

【0066】

まず、乱数列生成装置101の種受付部102は、wビットの整数の列 $s_1, s_2, \dots, s_n, \dots, s_m$ を種として受け付ける（ステップS201）。ただし、 $1 \leq n \leq m-1$ である。

【0067】

典型的には、 $s_1, s_2, \dots, s_n, \dots, s_m$ は、乱数列生成装置が有するRAM(Random Access Memory)等のメモリに格納されるが、CPUが有するキャッシュに格納してもよいし、ハードディスク等の読み書き可能な外部記録媒体に一時的に記憶してもよい。

【0068】

ついで、初期化部103は、当該受け付けられた $s_1, s_2, \dots, s_n, \dots, s_m$ を、整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として変換部104に与える（ステップS202）。

【0069】

ここで、 $x_1, x_2, \dots, x_n, \dots, x_m$ も同様に、RAM等のメモリに格納される。この場合、初期化部103が実行する処理は、 $s_1, s_2, \dots, s_n, \dots, s_m$ に対応するメモリから $x_1, x_2, \dots, x_n, \dots, x_m$ に対応するメモリへの値の転送によって実現することができる。

【0070】

さらに、変換部104は、当該与えられた整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ のそれぞれに対して上記の非線型変換 $g(\cdot, \cdot)$ により定義される変換を施してwビットの整数の列 $y_1, y_2, \dots, y_n, \dots, y_m$ を得る（ステップS203）。

【0071】

当該変換としては以下のような漸化式により定義されるものを採用することができる。

【0072】

(1) 整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(x_m, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う。

【0073】

(2) 所定の整数 c を用いて、整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(y_i, x_{i+1})$$

により変換を行う。

【0074】

(3) 整数 i ($1 \leq i \leq m-1$)についての漸化式

$$y_1 = g(c, x_1);$$

$$y_{i+1} = g(x_i, x_{i+1})$$

により変換を行う。

【0075】

これらの計算は、CPUが有するALU (Arithmetic Logic Unit) を用いて実現することができる。 $y_1, y_2, \dots, y_n, \dots, y_m$ もまた、メモリ等に格納される。

【0076】

そして、回転部105は、当該整数の列 y_{n+1}, \dots, y_m から回転ビット数を得る（ステップS204）。回転ビット数を得る手法としては、以下のようなものが考えられる。

【0077】

すなわち、 y_{n+1}, \dots, y_m をビット列と見て、あらかじめ定めたビット位置のビットを順に並べ、再度これを整数値と見る。図3に、 $w=4, m-n=2$ の場合に、所定のビット位置の値から整数値を得る様子を示す。本図の例では、8ビットのビット列から3ビット分を抽出している。

【0078】

3ビットから得られる整数値は、符号無し整数と見れば、0~7の8通りである。この場合は、回転方向として「あらかじめ定めた方向（右もしくは左）」を採用し、得られる整数値をそのまま回転ビット数とする。

【0079】

一方、1ビットを符号（正または負に対応付ける）とし、他の2ビットから回転量を得ることとして、この場合は、正の場合は左方向へ、負の場合は右方向へ、それぞれ絶対値分のビット数だけ回転するとしても良い。

【0080】

ついで、回転部105は、当該 $y_1, y_2, \dots, y_n, \dots, y_m$ を w_m ビットのビット列と見て、これ、もしくは、これの一部に対して所定の回転演算を行って、得られた w_m ビットのビット列から w ビットの整数の列 $z_1, z_2, \dots, z_n, \dots, z_m$ を得る（ステップS205）。

【0081】

所定の回転演算としては、以下のようなものを採用することができる。

【0082】

(1) w_n ビットのビット列を、得られた回転ビット数だけ巡回シフトするもの。図4に、この巡回シフトとして、 $w=4, n=4$ であって、 y_1, y_2, \dots, y_4 をビッグエンディアンに配列して、1ビット左にシフトする場合の概要構成を示した。これは、 w_m ビットのビット列の一部を回転する回転演算である。

【0083】

(2) w_m ビットのビット列全体を、得られた回転ビット数だけ巡回シフトするもの。図4における w_n ビットのビット列の回転と同様に、 w_m ビットのビット列全体を巡回させれば良い。

【0084】

(3) y_1, \dots, y_n の wn ビットのビット列を、得られた回転ビット数だけ巡回シフトするほか、これとは別に、 y_{n+1}, \dots, y_m の $w(m-n)$ ビットのビット列を、得られた回転ビット数だけ巡回シフトするもの。

【0085】

これらの手法は、メモリ等に格納された $y_1, y_2, \dots, y_n, \dots, y_m$ の全体もしくは一部を、CPUにとって自然なビット幅単位で、桁上がり・桁下がりを考慮しつつ、順次巡回シフトすることによって実現することができる。この場合、得られる $z_1, z_2, \dots, z_n, \dots, z_m$ は、 $y_1, y_2, \dots, y_n, \dots, y_m$ が格納されていたメモリ内の領域に新たな値として格納されることとなる。

【0086】

さらに、出力部 107 は、変換部 104 における変換ならびに回転部 105 における回転が、所定の回数繰り返されたか否かを判定する（ステップ S206）。

【0087】

たとえば、ステップ S201 の前において、メモリ内に用意されたカウンタ変数に「所定の回数の値」を代入し、ステップ S204 とステップ S206 の間において当該カウンタ変数の値を 1 減じ、ステップ S206 においては、当該カウンタ変数の値が 0 になったか否かを判定することによって、実現することができる。

【0088】

所定の回数繰り返された場合（ステップ S206；Yes）最後に得られた z_1, z_2, \dots, z_n を、乱数列 r_1, r_2, \dots, r_n として出力して（ステップ S207）、乱数列の生成を終了する。

【0089】

一方、所定の回数繰り返されていない場合（ステップ S206；No）、更新部 106 は、当該 $z_1, z_2, \dots, z_n, \dots, z_m$ を整数列 $x_1, x_2, \dots, x_n, \dots, x_m$ として変換部 104 に与えて（ステップ S208）、ステップ S203 に戻り、変換（ステップ S203）、回転（ステップ S204、S205）の処理を繰り返す。

【0090】

これは、 $z_1, z_2, \dots, z_n, \dots, z_m$ が格納されているメモリ等内の値を、 $x_1, x_2, \dots, x_n, \dots, x_m$ が格納されているメモリ等へ転送することによって実現できる。

【0091】

乱数列生成装置101においては、図示しない記憶部が存在し、その記憶部は、 $s_1, s_2, \dots, s_n, \dots, s_m, x_1, x_2, \dots, x_n, \dots, x_m, y_1, y_2, \dots, y_n, \dots, y_m, z_1, z_2, \dots, z_n, \dots, z_m, r_1, r_2, \dots, r_n, \dots, r_m$ などを異なる領域に、もしくは、値の利用の依存関係を分析することにより、同じ領域に記憶する（たとえば、 $y_1, y_2, \dots, y_n, \dots, y_m$ と $z_1, z_2, \dots, z_n, \dots, z_m$ 等。）ような構成をとることができる。また、各部は、上記の共有メモリを用いて互いに計算した値をやりとりするのである。

【0092】

図5は、本実施形態の乱数列生成装置101が実現されるコンピュータの典型的な概要構成を示す模式図である。以下、本図を参照して説明する。

【0093】

コンピュータ301は、CPU 302によって制御される。コンピュータ301に電源が投入されると、CPU 302は、ROM (Read Only Memory) 303に用意されたIPL (Initial Program Loader) を実行する。

【0094】

そして、IPLの実行により、フレキシブルディスクドライブ304に装着されたフレキシブルディスクやハードディスク305等に記録されたOS (Operating System) がロードされ、ユーザからの各種の指示入を受け付けることができるようになる。

【0095】

ユーザは、キーボード306やマウス307を操作して、コンピュータ301に対して各種の指示入力を与える。

【0096】

これに応じて、OSは、ハードディスク305やCD-ROM (Compact Disk ROM) ドライブ308に装着されたCD-ROMに記録されたプログラムや各種のデータをCPU 302に実行させ、実行の過程やその結果をディスプレイ3

09に表示する。

【0097】

また、CPU 302は、一時的な記憶域として、RAM 311を利用する。RAM 311は、上記のように、計算の途中で利用される各種の数列を記憶するために用いられる。

【0098】

さらに、CPU 302は、プログラムの実行の過程において、ハードディスク305に、生成された乱数列などの処理の結果や途中経過などの情報を保存することができる。

【0099】

なお、本実施形態における演算は、上記のように、きわめて単純なビット演算に還元することができる。したがって、専用の電子回路（加算器、減算器、シフタ、ラッチ等）を組み合わせて乱数列生成装置101を構成することができるほか、ASIC (Application Specific Integrated Circuit)、DSP (Digital Signal Processor) やFPGA (Field Programmable Gate Array) などのような電子回路の構成状態を可変に変更できる電子素子を利用して、乱数列生成装置101を構成することができ、これらの態様も本発明の範囲に含まれる。

【0100】

(実験の結果)

上記実施形態に係る乱数列生成装置101を用いて、以下の諸元で乱数列を生成させた。

$$W = 32,$$

$$n = 32,$$

$$g(a, b) = 2b^2 + h(a)b$$

【0101】

ただし、写像 $h(\cdot)$ は、与えられた値の数値表現の最下位2ビットを01に置換する演算により定義される。

【0102】

また、変換および回転は、それぞれ1ラウンドごとに1回とした。すなわち、

「所定の回数の繰り返し」は1回である。

【0103】

出力されるのは、全部で $wn = 1024$ ビットの乱数列 $r_1, r_2, \dots, r_{1024}$ である。

【0104】

これに対して 20000×89999 種類の種を与える、 20000×89999 ラウンドだけ乱数列 $r_1, r_2, \dots, r_{1024}$ を出力させた。

【0105】

そして、これに対して乱数列のランダム性を検査する標準的なテストである FIPS 140-1 ならびに FIPS 140-2 のうち、標準セキュリティ規格にあるランダム性検査テストを、1024 ビットの乱数列の中の各ビット位置毎に適用して、本実施形態の乱数列の性質を検査した。

【0106】

これらのテストにおいては、各ビット位置から 20000 ビットのビット列を取り出し、その 20000 ビットのビット列に対して、以下が行われる。

モノビットテスト (monobit test)。所定の位置のビットの値の出現頻度に偏りがないか否かを調べるもの。

ポーカーテスト (poker test)。20000 ビットを 5000 個の 4 ビットパターンに分割し、その 4 ビットパターンの出現頻度に偏りがないか否かを調べるもの。

ランズテスト (runs test)。乱数列から所定の長さの連を切り出した場合に、当該長さの連の出現頻度に偏りがないか否かを調べるもの。長さとしては 1 ~ 6 を用いる。

ロングランズテスト (long runs test)。ランズテストと同様であるが、FIPS 140-1 の場合は 34 以上の連が存在する場合ランダム性が否定され、FIPS 140-1 の場合は 26 以上の連が存在する場合ランダム性が否定される。

【0107】

実験の結果、FIPS 140-1においては、生成された 1024 ビット \times 89999 サンプルの 20000 ビット列は、すべて、定められた基準をクリア

した。

【0108】

また、FIPS 140-2においては、生成された1024ビット×89999サンプルの20000ビット列の内、99.92パーセントのサンプルが、定められた基準をクリアした。

【0109】

さらに、上記の乱数テストよりも厳しい乱数テストであるNIST 800-22に本発明を適用して調べた結果、回転の手法（3）を利用すると、極めて良好な乱数を得られることが判明している。

【0110】

また、本技術をXILINX（登録商標）社のVertex xcv1000（システムゲート数は100万）のFPGAに実装したところ、本アルゴリズムの並列性から、25.62Gビット/secのスピードで乱数列を生成させることができた。すなわち、本技術をFPGA等のハードウェアに実装すると、高速性の上で多大なメリットを得ることができる。

【0111】

このようにして、本実施形態によって生成された乱数列は極めて性質が良いものであり、秘密通信の暗号化の分野や物理現象、化学現象などの模擬実験の分野で有用であり、ハードウェア上で高速に良好なランダム性を持つ乱数列を出力するのに極めて有用であることが示された。

【0112】

（暗号化復号化装置）

上記の乱数生成装置を利用すれば、暗号化や復号化を実現することができる。図6は、このような暗号化や復号化を行う暗号化装置および復号化装置の概要構成を示す模式図である。

【0113】

暗号化装置601と、復号化装置651と、は、 $s_1, \dots, s_n, \dots, s_m$ を共通鍵とする。そして、それぞれが備える生成部602、652は、いずれも、同じ構成（計算手法）による乱数発生装置201を有し、この共通鍵 $s_1, \dots, s_n, \dots, s_m$ を入力

として受け付ける。すると、同じ乱数 r_1, \dots, r_n が out されることとなる。

【0114】

この乱数を用いて、暗号化装置601においては、メッセージ受付部603が受け付けた伝送メッセージの整数列 $p_1, p_2, \dots, p_i, \dots$ を、XOR部604が、 $p_1 \text{ xor } r_1, p_2 \text{ xor } r_2, \dots, p_i \text{ xor } r((i+n-1) \bmod n) + 1, \dots$ のように変換して、これを暗号化済メッセージの整数列 $e_1, e_2, \dots, e_i, \dots$ として出力する。

【0115】

ここで、 xor は上述したように排他的論理和演算を意味し、 $a \bmod n$ は a を n で割った余りを意味する。

【0116】

一方、復号化装置651のメッセージ受付部653は、暗号化済メッセージの整数列 $e_1, e_2, \dots, e_i, \dots$ を受け付け、XOR部654が $e_1 \text{ xor } r_1, e_2 \text{ xor } r_2, \dots, e_i \text{ xor } r((i+n-1) \bmod n) + 1, \dots$ のように変換して、これを復号化済メッセージの整数列 $f_1, f_2, \dots, f_i, \dots$ として出力する。

【0117】

ここで、

$$\begin{aligned} f_i &= e_i \text{ xor } r((i+n-1) \bmod n) + 1 \\ &= (p_i \text{ xor } r((i+n-1) \bmod n) + 1) \text{ xor } r((i+n-1) \bmod n) + 1 \\ &= p_i \end{aligned}$$

であるから、復号化済メッセージの整数列とともに伝送メッセージの整数列とは同じものであり、暗号化ならびに復号化ができることが示された。

【0118】

なお、処理の対象となるメッセージの長さは、 n 以下とすることもできる。その場合には、上記の $((i+n-1) \bmod n) + 1$ は i に置き換えることができる。このようにすれば、同じ乱数列を繰り返し利用するよりも、さらに秘匿性を高めることができる。

【0119】

上記のように、暗号化装置601と復号化装置651とは全く同じ構成であるので、1つの装置をある場合は暗号化装置601として、ある場合は復号化装置

651として、それぞれ利用することができる。

【0120】

【発明の効果】

以上説明したように、本発明によれば、生成される乱数列が乱数列として好ましい性質を有するような乱数列生成装置、乱数列生成方法、これらを用いた暗号化復号化装置、暗号化復号化方法、ならびに、これらをコンピュータによって実現するためのプログラムを提供することができる。

【図面の簡単な説明】

【図1】

本発明の実施形態に係る乱数列生成装置の概要構成を示す模式図である。

【図2】

本実施形態の乱数列生成装置において実行される乱数列生成処理の制御の流れを示すフローチャートである。

【図3】

本実施形態の乱数列生成装置の回転部において実行される回転ビット数の取得の様子を示す説明図である。

【図4】

本実施形態の乱数列生成装置の回転部において実行される回転演算の様子を示す説明図である。

【図5】

本実施形態の乱数列生成装置が実現される典型的なコンピュータの概要構成を示す模式図である。

【図6】

暗号化装置と復号化装置の実施形態の概要構成を示す模式図である。

【符号の説明】

101 乱数列生成装置

102 種受付部

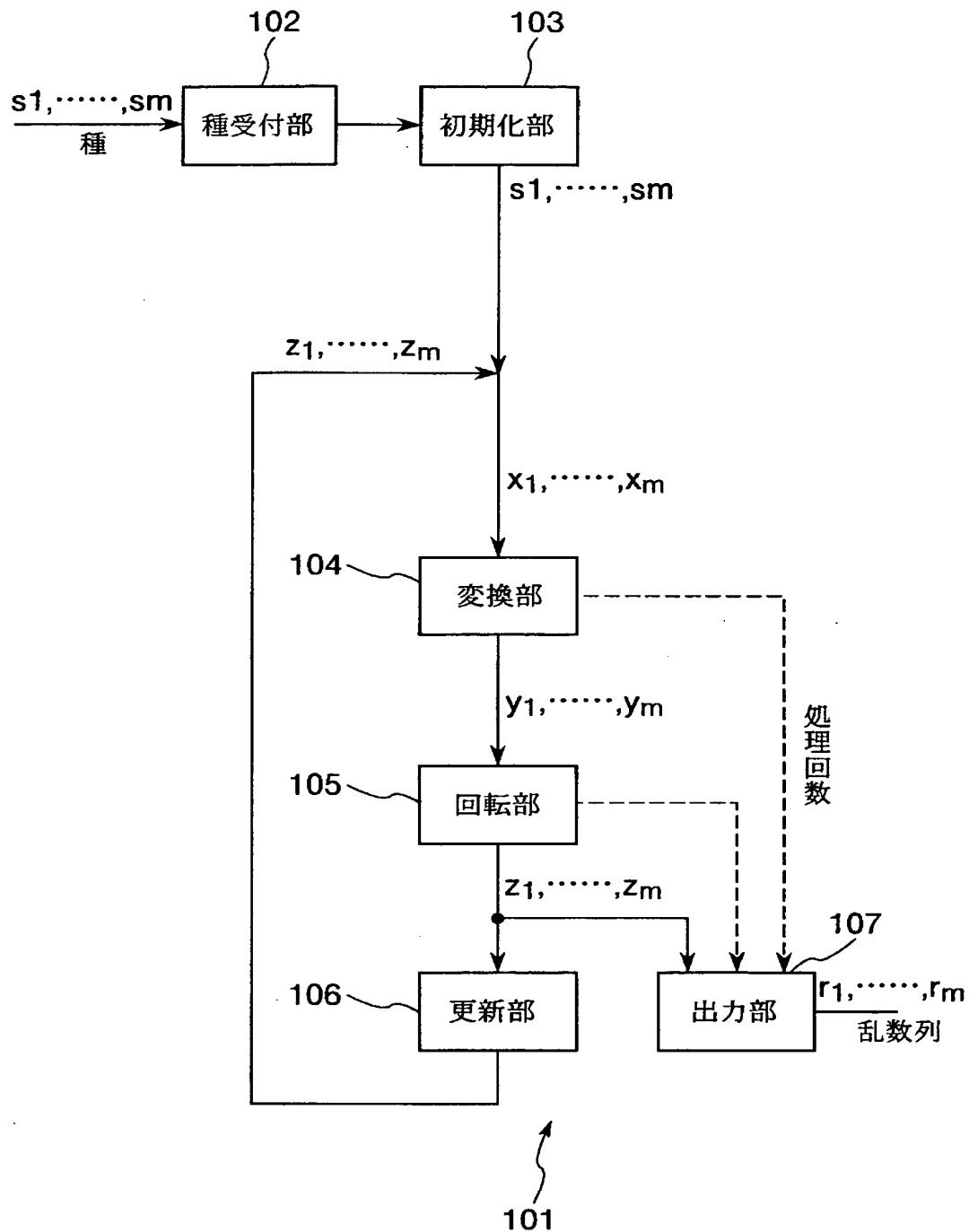
103 初期化部

104 変換部

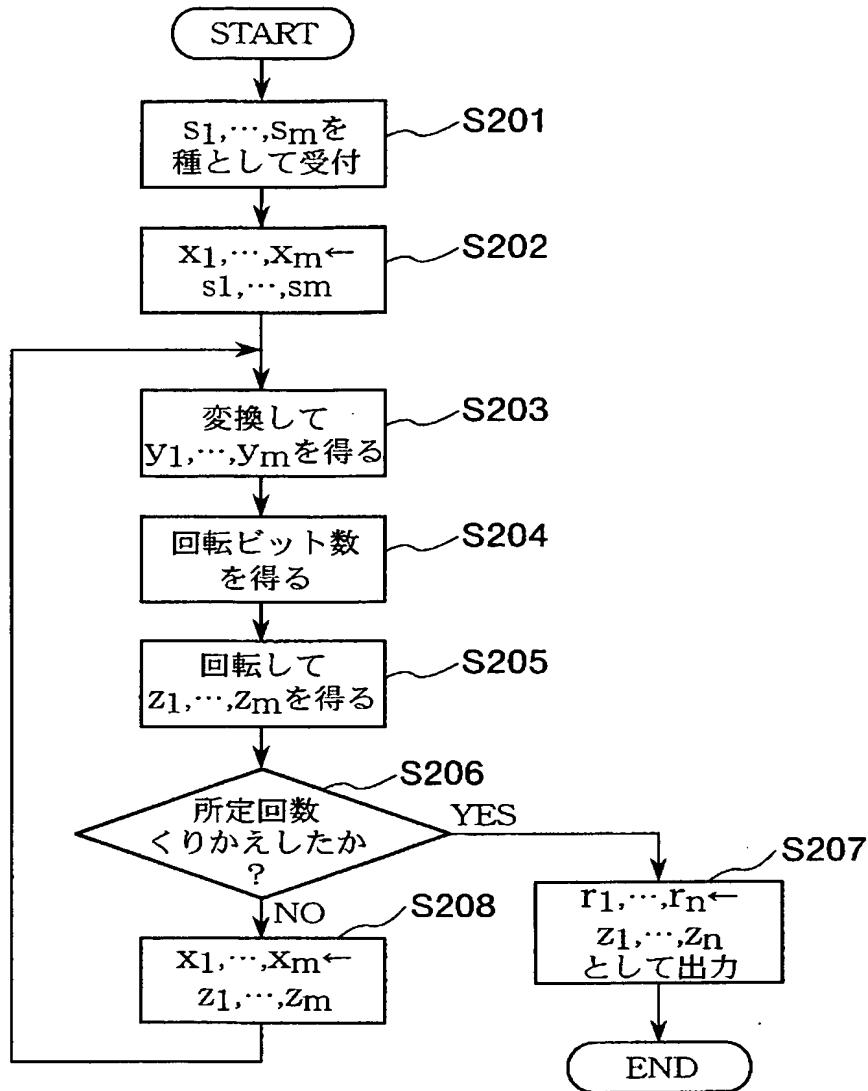
105 回転部
106 更新部
107 出力部
301 コンピュータ
302 C P U
303 R O M
304 フレキシブルディスクドライブ
305 ハードディスク
306 キーボード
307 マウス
308 C D - R O M ドライブ
309 ディスプレイ
311 R A M
601 暗号化装置
602 亂数生成部
603 メッセージ受付部
604 X O R 部
651 復号化装置
652 亂数生成部
653 メッセージ受付部
654 X O R 部

【書類名】図面

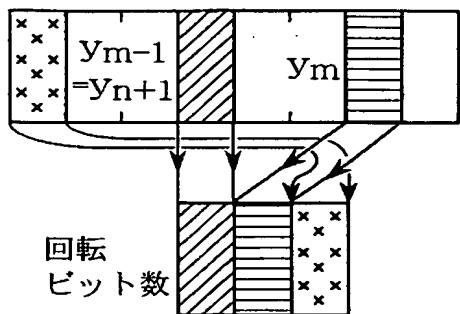
【図1】



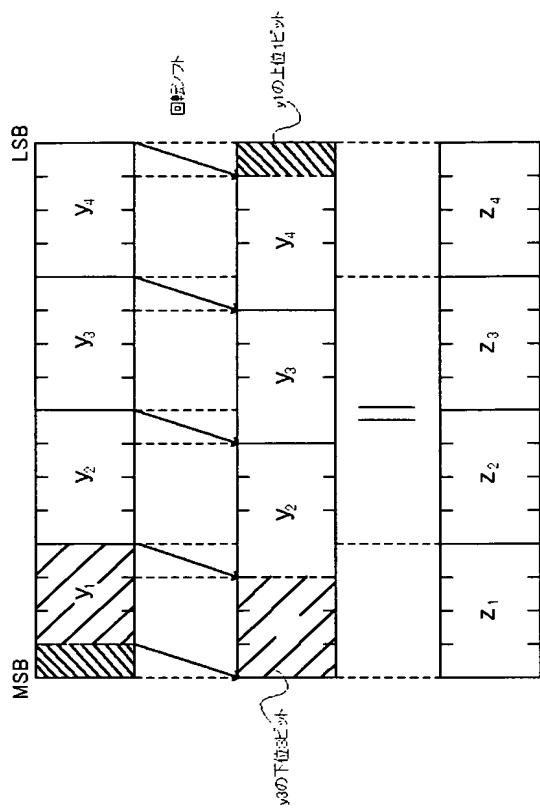
【図2】



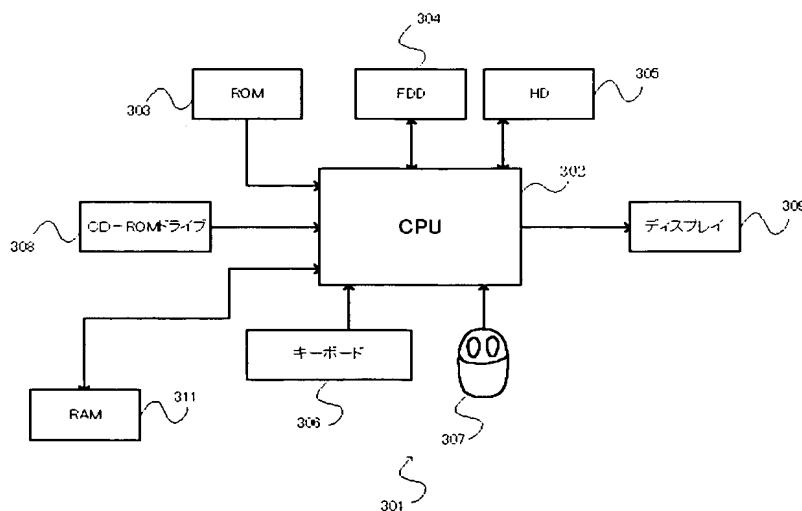
【図3】



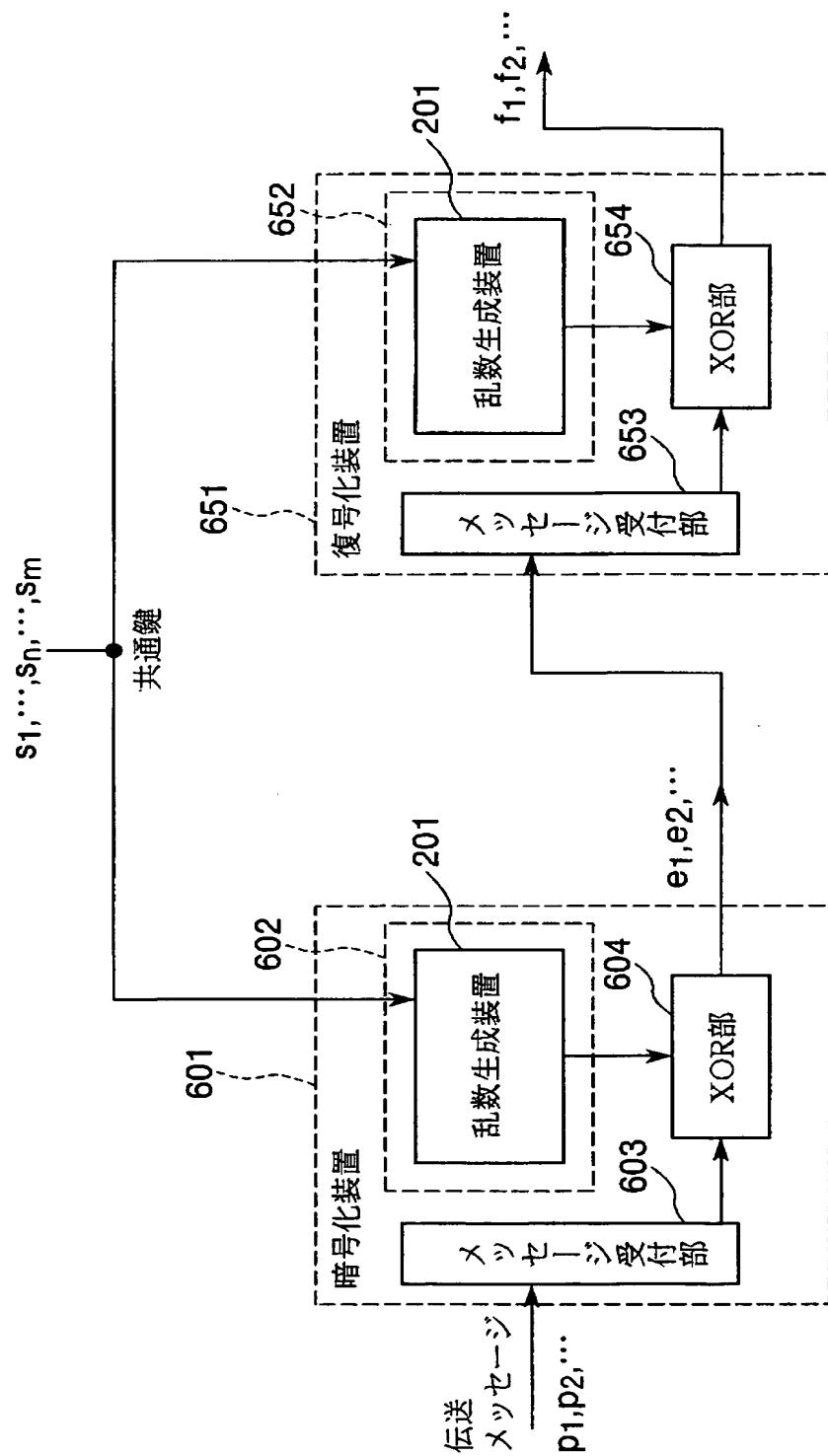
【図4】



【図5】



【図6】



【書類名】 要約書

【要約】

【課題】 亂数列生成装置等を提供する。

【解決手段】 亂数列生成装置 101 の種受付部 102 は、w ビットの整数の列を種として受け付け、初期化部 103 は、当該受け付けられた整数の列を、変換部 104 に与え、変換部 104 は、当該与えられた整数列のそれぞれに対して所定の変換を施して w ビットの整数の列を得て、回転部 105 は、得られた整数の列の一部から回転ビット数を得て、当該得られた整数の列をビット列と見て、これ、もしくは、この一部に対して当該得られた回転ビット数の回転演算を行つて、w ビットの整数の列を得て、更新部は、当該整数の列を変換部 104 に与え、出力部 107 は、変換部における変換ならびに回転部における回転が、所定の回数繰り返された場合、最後に得られた整数列の一部を乱数列として出力する。

【選択図】 図 1

特願 2003-075438

出願人履歴情報

識別番号 [301022471]

1. 変更年月日 2001年 4月 2日

[変更理由] 新規登録

住所 東京都小金井市貫井北町4-2-1
氏名 独立行政法人通信総合研究所